

Self-Supervised On-Policy Distillation for Omni-Modal Video Captioning

Zinuo Li¹, Yongxin Guo², Xin Li³, Hao Tan³, Zhao Zhong³,
Liefeng Bo³, Mohammed Bennamoun¹, Farid Boussaid¹, QiuHong Ke^{4*}

¹ University of Western Australia ² Chinese University of Hong Kong (SZ)

³ Tencent ⁴ Monash University

Abstract

Video captioning directly reflects a Video Large Language Models perception, making caption fidelity a strong proxy for audio-visual grounding. Existing approaches often rely on costly annotations or stronger external teachers, limiting scalability. In this paper, we instead explore whether a VideoLLM can improve its own captions by exploiting structure already present within the model through self-supervision. Our exploration builds on two asymmetries we observe in current VideoLLMs. The first is a task asymmetry: captioning requires broad open-ended perception, whereas question answering narrows reasoning to specific dimensions. The second is an information asymmetry: the same model reasons more reliably with richer context or higher-fidelity audio-visual inputs. Motivated by these observations, we propose *SODA (Self-Supervised On-Policy Distillation via Dual-Stream Auditing)*, a framework that enables a VideoLLM to critique and refine its own captions. A student first produces an initial caption from a low-budget video view, followed by a dual-stream audit: a *Caption QA Stream* identifies factual errors, and a *Video QA Stream* detects omitted content. Conditioned on this Revision QA Bank, a teacher using a larger input budget rewrites the caption, and the refined outputs are distilled back into the student. Experiments suggest that, despite training only on the captioning objective, SODA improves caption fidelity, reduces hallucination, and remains competitive on general video understanding.

1 Introduction

Improving captioning fidelity of Video Large Language Models (VideoLLMs) typically relies on either finer-grained multimodal annotations or stronger external teacher models (Chen et al., 2024a; Zhang et al., 2024b). Both are costly, hard

to scale, and often brittle. Fine-grained video annotation is expensive and often emphasizes salient events rather than the fine-grained perceptual details that VideoLLMs frequently miss (Zhang et al., 2025c; Wang et al., 2024), leading to misaligned supervision. Meanwhile, teacher models trained through supervised fine-tuning or reinforcement learning methods such as RLHF and GRPO (Schulman et al., 2017; Shao et al., 2024) typically provide supervision only through final responses, offering limited guidance for token-level supervision.

One approach to overcoming this limitation is On-Policy Distillation (OPD) (Agarwal et al., 2024; Ko et al., 2024), which provides denser supervision by aligning student and teacher outputs at the token level along the students own trajectories. However, OPD still depends on stronger external teachers, whose training for omni-modal video understanding is itself expensive. We therefore ask a more modest question: can the model itself provide useful structure to guide its own captioning training, providing additional signal in a self-supervised manner? Rather than relying on stronger teachers as in OPD, we explore whether a low-resource, low-annotation approach that requires neither external teachers nor additional human labels can provide sufficient supervision on its own. Our approach is motivated by two asymmetries observed in current VideoLLMs as follows.

The first is a *task asymmetry*: the captioning task is required to jointly cover all perceptual dimensions in a single open-ended generation, whereas question answering already narrows reasoning to specific dimensions. From the model’s perspective, this difference manifests in attention: a question implicitly specifies what the model should attend to, whereas captioning requires the model to determine both where to attend and what to describe without an explicit query to

*Corresponding author.

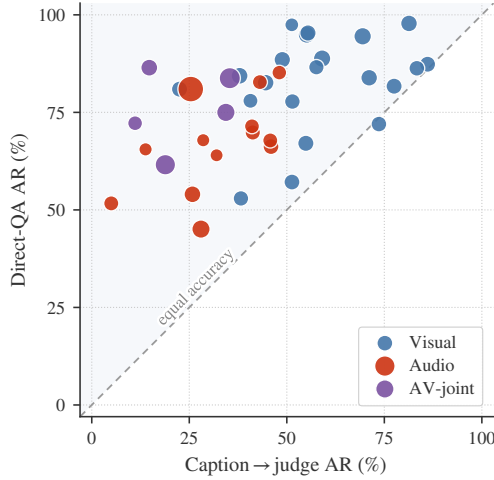


Figure 1: QA accuracy versus caption accuracy on each VCapsBench-Omni dimension. A point above the diagonal means the model answers the question more accurately from the video than from its own caption; color indicates the modality and larger dots cover more questions. All dimensions but one lie above the line, so the model already knows more than it writes, and the gap is largest on audio and audio-visual content, where the caption often stays silent.

guide attention (Dai et al., 2026). This difficulty gap is what allows question answering to audit captioning: judging whether a caption is faithful does not require writing a better one; instead, it only requires sampling questions from the caption and checking whether the answers generated from the video are consistent with the captions claims. Question answering can therefore provide supervision for captioning. Figure 1 and Table 1 show that this gap is real and large: holding the model and the questions fixed, answering a question directly from the video is more accurate than reading the same answer off the model’s own caption, and the widest gaps fall on audio and audio-visual content, which is exactly where free-form captions tend to stay silent. Appendix A.2 gives the per-modality breakdown and the full protocol.

	AR↑	IR↓	CR↑
caption→judge	43.5	26.3	59.1
direct-QA	77.3	14.2	90.1
Δ	+33.8	-12.1	+31.0

Table 1: Overall caption-vs-QA probe on the base model. Direct-QA answers each question from the video; caption→judge reads the answer from the model’s own caption. Per-modality numbers are in Appendix A.2.

The second is an *information asymmetry*: even with model parameters fixed, reasoning reliability improves when the model is provided with

richer textual and multimodal information (Klissarov et al., 2026; Sang et al., 2026). On the textual side, providing the models initial caption together with self-generated Revision QA pairs identifying potentially incorrect or incomplete parts of the caption improves reasoning performance, as feedback generated by the model itself has been shown to improve reasoning in prior work. On the multimodal side, improving the fidelity of the same video and audio inputs through more sampled frames, higher spatial resolution, or longer audio context also improves question-answering accuracy, with higher spatial resolution in particular consistently benefiting video understanding benchmarks. We refer to these as two *information lifts*: both improve reasoning reliability without modifying model parameters, operating purely through richer inputs.

These two asymmetries motivate the design of our method *Self-Supervised On-Policy Distillation via Dual-Stream Auditing* (SODA). The information asymmetry motivates a teacher-student setup in which the same model operating under a richer input budget serves as the teacher, while the same model operating under a tighter input budget serves as the student. Because the teacher and student share parameters, the resulting supervision arises from differences in available information rather than model capacity. The task asymmetry chooses the curriculum: the model audits its own draft with two QA streams, one seeded from the caption (precision) and one from the audio-visual input (recall) (Wang et al., 2023); the teacher rewrites the initial caption conditioned on the resulting evidence. The student is then trained to match the teachers revised caption distribution by minimizing a token-level KullbackLeibler divergence between their output distributions, providing dense per-token supervision beyond answer-level feedback. Because the student is trained under constrained input budgets, the distilled model tends to retain captioning quality and downstream performance even when the test-time input budget is reduced, improving token-budget efficiency for deployment.

Our contributions can be summarized in three points:

- We explore the feasibility of self-distillation for video captioning without an external teacher or additional human labels by leveraging two key asymmetries in VideoLLMs:

a task asymmetry that lets the model’s own question answering audit its own captions, and an input-budget asymmetry that lets the same model reason more reliably under richer input.

- We propose SODA, a self-supervised, self-evolving distillation framework that enables a VideoLLM to critique and refine its own captions. SODA uses dual-stream QA auditing to identify incorrect and missing content in the output caption, then distills the revised captions back into the model through dense token-level supervision.
- We introduce *VCapsBench-Omni*, a carefully curated subset of VCapsBench (Zhang et al., 2025c) with audio and audio-visual joint dimensions, enabling a comprehensive evaluation of caption fidelity across visual, audio, and joint audio-visual dimensions.

2 Related Works

SODA relates to three lines of work. *Classical KD and on-policy distillation* (Hinton et al., 2015; Kim and Rush, 2016; Wen et al., 2023; Gu et al., 2024; Agarwal et al., 2024; Ko et al., 2024; Yang et al., 2026) relies on a teacher that is stronger than the student in parameters, data, or compute. *Self-distillation and self-training* (Furlanello et al., 2018; Zhang et al., 2019; Tarvainen and Valpola, 2017; Xie et al., 2020; Zelikman et al., 2022; Singh et al., 2024; Wang et al., 2023; Zohar et al., 2025) typically separates teacher and student along a noise axis and supervises on the final answer. *Hallucination mitigation for multimodal LLMs* (Li et al., 2023; Guan et al., 2024; Wang et al., 2024; Li et al., 2025a; Zhang et al., 2025b; Sung-Bin et al., 2025; Leng et al., 2024; Huang et al., 2024; Yu et al., 2024; Zhou et al., 2024a; Zhao et al., 2024; Zhou et al., 2024b; Rafailov et al., 2023; Wang et al., 2026; Li et al., 2025b; Shao et al., 2024; Schulman et al., 2017; Wang et al., 2025a) relies on curated preference data, reward models, or an RL loop on top of answer tokens. SODA differs from all three: a weight-shared teacher whose edge is informational rather than capacity-based, a dense token-level KL on caption tokens that provides per-token soft targets, and no external annotation or reward signal. We expand each thread in Appendix A.9.

3 Methodology

3.1 Overview

Inspired by the task asymmetry between captioning and question answering, and the information asymmetry induced by richer inputs, we propose *Self-Supervised On-Policy Distillation via Dual-Stream Auditing (SODA)*, a self-distillation framework in which a single VideoLLM serves as both the student and teacher for its own captioning process. SODA retains the dense per-token supervision of on-policy distillation while replacing stronger external teachers with self-generated QA-based auditing signals, enabling training in low-resource settings without additional annotations. ***The core idea of SODA is to create and then distill an information gap within the same model.*** The teacher branch receives both (1) richer audio-visual inputs and (2) QA-based feedback on the student’s initial caption, allowing it to generate more reliable revised captions. These revised outputs are then distilled back into the student, enabling the same model to retain strong performance even under constrained inference-time input budgets.

As illustrated in Figure 2, training alternates two phases. In Phase A, the model first generates an initial caption, then audits it through two QA streams using richer audio-visual inputs, and finally revises the caption based on the resulting evidence. This produces a revised caption c_r that captures the additional information available to the teacher in textual form. In Phase B, this additional information is distilled back into the student. Both teacher and student are trained with the revised caption c_r as the target, and training minimizes a token-level KL divergence between their caption distributions. Because c_r is generated using richer inputs and QA-based auditing, the resulting dense per-token targets provide substantially more supervision than answer-level feedback alone. Importantly, both the auditing process and the richer input budget are used only during training, leaving inference-time computation unchanged.

3.2 Phase A: Caption Revision

Phase A enables the model to audit and revise its initial caption through QA-based self-reflection. The phase consists of three steps, each producing one component of a per-video tuple: (1) *drafting* yields the student’s initial caption c_s ; (2) *dual-stream auditing* yields the Revision QA Bank $\mathcal{E}_{\text{rgb}}$, a set of model-produced question and answer pairs

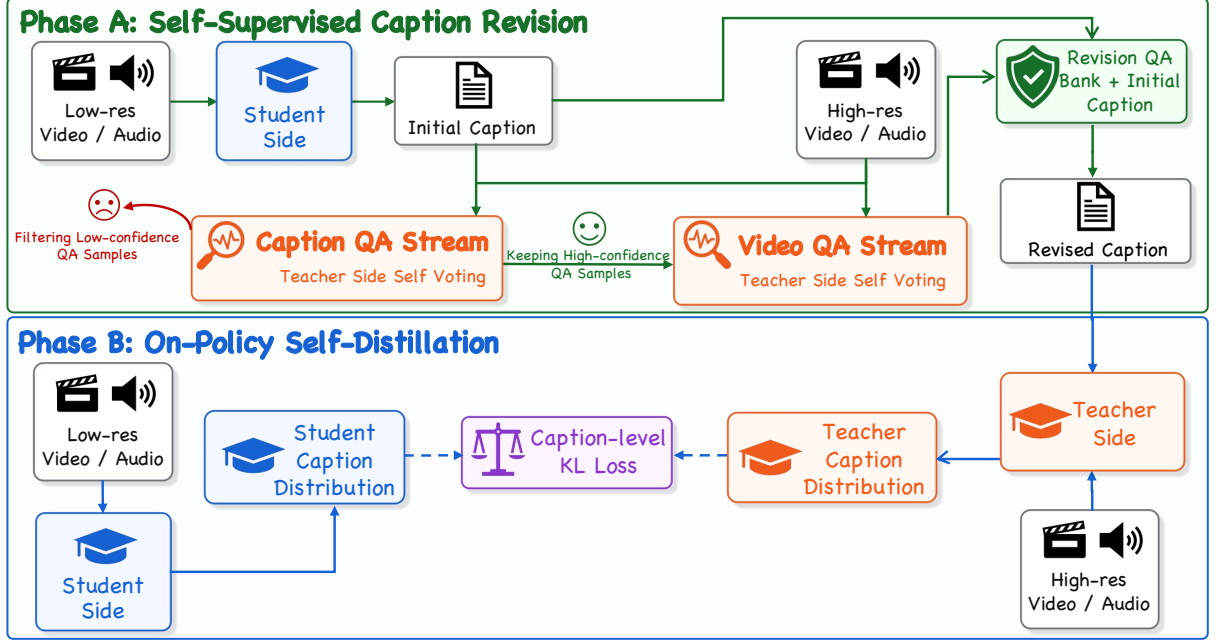


Figure 2: Overview of SODA. Phase A: the same model produces an initial caption c_s from the low-budget view V_{lo} , audits it with the *Caption QA Stream* (questions from c_s , catching commissions) and the *Video QA Stream* (questions from the high-budget view (V_{hi}, A_{hi}) , catching omissions); K -sample self-consistency voting yields the Revision QA Bank $\mathcal{E}_{rqb}$, and the teacher revises c_s into a revised caption c_r . Phase B: both roles are teacher-forced to c_r ; we minimize a token-level KL between the student logits (conditioned on V_{lo} and prompt_c) and the teacher logits (conditioned on (V_{hi}, A_{hi}) , c_s , $\mathcal{E}_{rqb}$), with stop-gradient on the teacher side. The audit pipeline and the high-budget view are training-only.

that catch what c_s gets wrong or misses; and (3) *revision* yields the revised caption c_r obtained by rewriting c_s based on $\mathcal{E}_{rqb}$. The resulting tuple $(c_s, \mathcal{E}_{rqb}, c_r)$ is cached and consumed by Phase B.

Initial Caption. Let M_θ denote the VideoLLM with parameters θ , and let $M_\theta(\cdot \mid \text{ctx})$ denote its autoregressive distribution over output tokens given a context. Given a video with visual and audio streams (V, A) and a fixed captioning instruction prompt_c , the student drafts an initial caption

$$c_s \sim M_\theta(\cdot \mid V, A, \text{prompt}_c). \quad (1)$$

This draft corresponds to the students standard inference-time output and serves as the target of the subsequent auditing process.

Dual-Stream Auditing. A faithful caption should capture a broad range of perceptual aspects of a video, but a free-form draft c_s often overlooks important details. To explicitly define the scope of faithful captioning, we adopt a taxonomy of $D=29$ audio-visual dimensions (Zhang et al., 2025c) spanning visual content (e.g. subjects, actions, spatial layout, fine details, scene segments), camera and aesthetic (camera movement, shot type, depth of field, color grading, visual style), on-screen text (native OCR, overlay text), audio

(speech content, voice attributes, audio events, sound scene, audio-visual sync), and cross-modal checks (anti-hallucination, content type); the full list is given in Appendix A.4.

A typical c_s omits many of these dimensions and may also contain uncertain or incorrect descriptions of the dimensions it does cover. The audit therefore serves two complementary purposes: verifying the claims made in c_s and recovering content that c_s omits or describes ambiguously. A single QA stream cannot reliably accomplish both goals. Questions derived from the caption cannot probe content that the caption never mentions, while questions derived from the video can recover missing content but are less effective at verifying specific claims already made in the caption. This motivates a *dual-stream auditing process* in which caption-grounded and video-grounded QA evidence are combined and filtered through self-consistency voting.

To this end, we provide the auditing process with richer inputs than those used to generate the initial caption. Specifically, we construct a high-budget version (V_{hi}, A_{hi}) of the same video, with more sampled frames, higher resolution, and a longer audio window, and run the student on a

tighter low-budget view V_{lo} . On top of this expanded view, every question q is annotated with a dimension index $d \in \{1, \dots, D\}$ and a modality $m \in \{\text{visual}, \text{audio}\}$. The two QA streams serve complementary roles and differ only in how their questions are generated:

- **Caption QA Stream** (*verifying* the claimed dimensions). For each dimension, M_θ either generates a verification question grounded in c_s , or returns SKIP_NOT_IN_CAPTION when c_s does not address that dimension. The skipped dimensions are collected in $S_{\text{skip}} \subseteq \{1, \dots, D\}$, while the remaining questions form a candidate pool whose voted answers populate $\mathcal{E}_{\text{rqb}}^{\text{cap}}$.
- **Video QA Stream** (*recovering* the missing dimensions). Let U be the union of S_{skip} and the dimensions whose Caption-stream answer is rejected or only weakly supported by the voting rule below (using a looser pivot threshold $\tau_{\text{piv}} \leq \tau$), optionally capped at D_{max} entries per video. On the dimensions in U , M_θ generates fresh questions directly from $(V_{\text{hi}}, A_{\text{hi}})$, while avoiding previously used CaptionQA questions. The resulting answers form a second pool that populates $\mathcal{E}_{\text{rqb}}^{\text{vid}}$.

The two streams thus return question *pools* but not yet trustworthy answers, since a single sample to an open-ended question may be unreliable. To obtain reliable revision signals, both streams use a shared **Self-Voting Mechanism**. For each q , the teacher samples $K = 5$ independent answers from the high-budget view,

$$a^{(k)} \sim M_\theta(\cdot \mid V_{\text{hi}}, A_{\text{hi}}, q), \quad k = 1, \dots, K, \quad (2)$$

under self-voting sampler (T, p) with temperature $T = 0.7$ and nucleus probability $p = 0.9$. Before voting, answers are normalized so that semantically equivalent responses count as the same answer. Specifically, we strip trailing parenthetical justifications, lowercase and collapse whitespace, and map hedging phrases (e.g., “not sure”, “cannot tell”, “unknown”) into a single “uncertain” bucket. The teacher then takes the majority answer $\hat{a}$ together with its agreement rate $\rho \in [0, 1]$, and retains $(q, \hat{a})$ only if $\rho \geq \tau$ and $\hat{a} \neq \text{“uncertain”}$, the latter dropping questions on which the model itself declines to answer. The agreement ρ thus serves as a self-calibrated reliability signal that filters out low-confidence or hedged questions before they enter the audit; the looser threshold τ_{piv}

referenced above reuses this same signal to flag which Caption-stream dimensions need to be re-asked by the Video stream. Merging the survivors of both streams gives $\mathcal{E}_{\text{rqb}} = \mathcal{E}_{\text{rqb}}^{\text{cap}} \sqcup \mathcal{E}_{\text{rqb}}^{\text{vid}}$ (disjoint union after re-indexing). This model-generated QA set serves as revision evidence for the next stage.

Revision. The audit tells the teacher *what* to fix; the revision step decides *how*. We pack c_s and $\mathcal{E}_{\text{rqb}}$ into a fixed revision-instruction template $\text{prompt}_r(c_s, \mathcal{E}_{\text{rqb}})$ that asks the model to rewrite c_s given $\mathcal{E}_{\text{rqb}}$, and the teacher rewrites the caption in a single greedy pass conditioned on the high-budget view,

$$c_r = \arg \max M_\theta(\cdot \mid V_{\text{hi}}, A_{\text{hi}}, \text{prompt}_r(c_s, \mathcal{E}_{\text{rqb}})). \quad (3)$$

The prompt treats each $\hat{a}_i$ as taking priority over c_s , and drops any audio claim whose entry has $\hat{a}_i = \text{none}$. Greedy decoding makes c_r a stable target for Phase B. We cache the tuple $(c_s, \mathcal{E}_{\text{rqb}}, c_r)$ for each training video and pass it on to Phase B.

3.3 Phase B: On-Policy Self-Distillation

Phase A produces c_r as a richer text target, but writing better captions to a cache is not yet learning. Phase B closes the loop by treating c_r as the bridge through which the teacher’s informational advantage is folded back into the student’s parameters. For each cached tuple, we run two forward passes, both teacher-forced to c_r :

$$\ell_s = M_\theta(V_{lo}, \text{prompt}_c; \text{forced}=c_r) \quad (4)$$

$$\ell_t = M_\theta(V_{\text{hi}}, A_{\text{hi}}, \text{prompt}_r(c_s, \mathcal{E}_{\text{rqb}}); \text{forced}=c_r) \quad (5)$$

where $\ell_s, \ell_t \in \mathbb{R}^{|c_r| \times |\mathcal{V}|}$ are per-token logits over the model vocabulary $\mathcal{V}$ at every caption position, and $|c_r|$ denotes the token length of c_r . The captioning prompt prompt_c and the revision prompt prompt_r are the same fixed templates used in Phase A; their full strings are given in Appendix A.6. The two paths differ by both information lifts: the teacher path conditions on c_s and $\mathcal{E}_{\text{rqb}}$ and reads the high-budget view, while the student path has neither, so the KL transfers the text-side and the multimodal lift together; Section 4.4 separates them. Only ℓ_s carries gradients. The loss is the average token-level KL over the caption span,

$$\mathcal{L}_{\text{SODA}}(\theta) = \frac{1}{|c_r|} \sum_{j=1}^{|c_r|} \text{KL} \left(\text{softmax}(\ell_s^{(j)}) \parallel \text{sg} \left(\text{softmax}(\ell_t^{(j)}) \right) \right), \quad (6)$$

where $\text{sg}(\cdot)$ is the stop-gradient operator. Intuitively, the student must recover the per-token distribution that the same model produces with c_s , $\mathcal{E}_{\text{rgb}}$, and the high-budget view, while itself seeing none of them; closing this gap forces the missing information to be encoded in the parameters rather than supplied at inference time. We optionally replace KL with Jensen-Shannon (mixture coefficient $\beta \in [0, 1]$), which we find slightly more stable early in training.

4 Experiments

4.1 Setup

We leave further details in Appendix A.1.

Backbone. We build SODA on Qwen2.5-Omni-7B (Qwen Team, 2025), an open-source omni-modal model that processes video, audio, and text in a single transformer backbone. We keep the vision encoder and the audio tower frozen and insert LoRA adapters on the language backbone; the same checkpoint plays both student and teacher.

Training data. We use 2,000 unlabeled videos sampled from Panda-70M (Chen et al., 2024b), a publicly released collection of short clips. Each video is processed once by Phase A to produce a cached tuple $(c_s, \mathcal{E}_{\text{rgb}}, c_r)$, with no human annotation and no overlap with any evaluation benchmark.

Caption-fidelity benchmarks. We evaluate captioning along three LLM-as-judge benchmarks. *Omni-Cloze* (Ma et al., 2026) fills modality-specific blanks inside the predicted caption and reports per-modality and overall accuracy. *VCapsBench-Omni* (our extension of VCapsBench (Zhang et al., 2025c); see Appendix A.5) reports accuracy rate (AR $\uparrow$), inconsistency rate (IR $\downarrow$), and coverage rate (CR $\uparrow$) per modality (visual, audio, audio-visual joint). *UGC-VideoCap* (Wu et al., 2025) scores user-generated-content captions along visual, audio, and detail sub-dimensions.

General video understanding benchmarks. We use *VideoHalluciner* (Wang et al., 2024), whose adversarial binary questions are paired into Basic (factual) and Hallucinated (counterfactual) on the same video, so that Overall requires both correct and is therefore sensitive to over-confident hallucination; and *Video-MME v2* (Fu et al., 2026), which grades questions at three reasoning levels with a non-linear scoring scheme and draws most of its videos from after 2025. SODA does not

directly optimize answer tokens on either, so improvements indicate caption-fidelity gains transferring to downstream general video understanding.

4.2 Main Results: Caption Fidelity

Table 2 reports caption-fidelity results across the three benchmarks, comparing SODA against a panel of recent open-source VideoLLMs at a comparable parameter scale (7B to 8B). Rather than narrating per-cell deltas, we draw three observations.

- **Where the audit pushes is where the model improves.** The visible gains over Qwen2.5-Omni-7B fall on the audio and detail axes (Omni-Cloze audio, UGC-VideoCap audio and detail), the same dimensions on which the Revision QA Bank most often overturns or supplements the initial caption; the visual axis moves the least and on UGC-VideoCap even regresses slightly.
- **Recall, not precision, is the bottleneck the audit unblocks.** On VCapsBench-Omni the gain on coverage rate (CR) is far larger than on accuracy rate (AR), and inconsistency rate (IR) drifts up slightly. The IR drift is not supervision noise: omission, not commission, is the dominant failure mode of free-form captions (Appendix Figure 5), and once the video-grounded stream forces the model to address dimensions it would have skipped, some borderline calls surface. The audit turns a silent miss into an explicit, gradeable claim, which is the prerequisite for correcting it later.
- **Per-token KL transfers granularity, not just coverage.** On UGC-VideoCap the largest lift is on the *Detail* sub-score, which is hard to inflate by simply enumerating more entities. This is consistent with Phase B teaching the student not only *what* to mention but at *what granularity*, since the teacher-forced c_r encodes both choices jointly token by token.

4.3 Main Results: General Video Understanding Transfer

SODA’s training loss touches only the captioning span; it never sees an MCQ answer token. Table 3 therefore reports the more pointed question of whether captioning-only self-supervision *costs* general video understanding. We highlight three observations.

Method	Omni-Cloze [†]				VCapsBench-Omni			UGC-VideoCap (Final Caption) [†]			
	Vis [↑]	Aud [↑]	AV [↑]	Tot [↑]	AR [↑]	IR [↓]	CR [↑]	Aud [↑]	Vis [↑]	Det [↑]	Avg [↑]
<i>Proprietary Omni-Modal Models</i>											
Gemini 2.0 Flash (Google DeepMind, 2025a)	32.3	31.7	40.1	33.2	-	-	-	-	-	-	-
Gemini 2.5 Flash (Google DeepMind, 2025b)	24.4	36.2	41.4	33.0	-	-	-	74.2	78.8	77.2	76.7
Gemini 2.5 Pro (Google DeepMind, 2025c)	40.8	44.1	52.8	43.6	62.1	15.4	73.5	70.8	75.8	74.8	73.8
<i>Open-Source Omni-Modal LLMs</i>											
video-SALMONN-13B (Sun et al., 2024)	3.5	2.3	4.6	3.3	-	-	-	-	-	-	-
VideoLLaMA2-7B (Cheng et al., 2024)	7.6	4.3	8.7	6.6	-	-	-	-	-	-	-
Gemma-3n-E2B-it (Google DeepMind, 2025d)	-	-	-	-	46.0	29.3	67.4	52.0	52.4	49.6	51.3
Gemma-3n-E4B-it (Google DeepMind, 2025d)	-	-	-	-	47.8	29.7	69.4	54.2	54.6	51.2	53.3
Qwen2.5-Omni-3B (Qwen Team, 2025)	-	-	-	-	37.6	28.5	52.6	48.2	55.6	52.6	52.2
MiniCPM-o-2.6-8B (OpenBMB, 2024)	-	-	-	-	30.6	29.7	41.7	46.0	59.7	<u>62.4</u>	56.3
video-SALMONN 2+ (Tang et al., 2025)	-	-	-	28.3	37.0	-	47.2	-	-	-	68.1
Qwen2.5-Omni-7B (base) (Qwen Team, 2025)	<u>18.3</u>	<u>14.1</u>	<u>21.9</u>	16.6	44.5	29.7	60.7	<u>57.6</u>	<u>59.4</u>	57.0	58.0
+ SODA (ours)	27.1	21.9	24.8	<u>25.0</u>	<u>47.7</u>	30.3	<u>68.4</u>	73.1	57.2	73.3	<u>67.8</u>

Table 2: Caption-fidelity results on Omni-Cloze (Visual / Audio / Audio-Visual / Total accuracy), VCapsBench-Omni (AR / IR / CR), and UGC-VideoCap (Final Caption sub-scores: Audio / Visual / Detail / Average). “-” marks that the metrics are not available on the corresponding models. Per column, the best open-source result is in **blue bold** and the second-best is underlined. [†] quoted from official papers for all rows but ours; VCapsBench-Omni is run by us for every model.

- **Captioning-only training even slightly improves general ability.** On VideoHalluciner the SODA-trained student improves on all three slices (Basic, Hallucinated, and the stricter Overall that requires both correct on the same video); on Video-MME v2 it is on par at Level 1 and slightly above the base at the harder Levels 2 and 3. This is the opposite of the side effect typically observed when a narrow objective is sharpened by SFT, where probes outside the loss often *regress* as the price of the in-task gain.
- **On-policy targets keep the supervision inside the model’s own support.** We attribute this benign side effect, tentatively, to the on-policy and self-supervised nature of the target. The revised caption c_r is greedily decoded by the same model from inputs it has already encoded, and Phase B’s teacher-forced KL only pulls the student toward a distribution that already lies within its own support: there is no foreign distribution to be *covered*, only a redistribution of mass within one that the model already represents. Conventional SFT on captions written by a different annotator, by contrast, asks the model to assign mass to sequences far from its prior, and the optimizer can satisfy that demand by overwriting capabilities that were not part of the loss. We do not claim a causal mechanism on the basis of two probes; what the experiments show is that the side effect of SODA on the general-purpose probes we measured

is, at minimum, not harmful.

- **Gains concentrate on the slices where audio-visual fidelity matters most.** The VideoHalluciner gain is larger on the *Hallucinated* slice (which penalizes over-claiming content not supported by the input) than on *Basic*, and largest on *Overall*; on Video-MME v2 the gains concentrate on Levels 2 and 3, the harder reasoning tiers. Both patterns are consistent with the audit specifically removing unsupported claims from c_r during training, so what transfers to MCQ time is not only a richer caption distribution but a calibrated reluctance to assert content the audio-visual evidence does not back.

4.4 Ablations

We report four ablations that target the key design choices of SODA, summarized in Table 4. The first row repeats the base Qwen2.5-Omni-7B as the Round 0 reference. Each row turns off or changes one part of the recipe. We start before training: *Revised Caption Only* (c_r) is the caption Phase A writes on the base model, with Phase B never run, so it shows what the audit alone can do. The next two rows take away one of the teacher’s two advantages each. *w/o Extra Vid/Aud Info* keeps the full audit but drops the teacher to the student’s low video and audio budget, so only the input-budget lift is gone. *w/o Audit* does the reverse: the teacher keeps the high budget but writes one caption straight away, with no questions asked. Having isolated what the teacher sees, the next two rows ask when its weights are read. *EMA update*

Method	VideoHalluciner [†]			Video-MME v2 (w/o subtitles) [†]		
	Basic [↑]	Hallu. [↑]	Overall [↑]	Level 1	Level 2	Level 3
<i>Proprietary Omni-Modal Models</i>						
GPT4-o (OpenAI, 2024)	75.1	74.2	53.3	-	-	-
<i>Open-Source VideoLLMs</i>						
Video-LLaMA2-7B (Cheng et al., 2024)	<u>90.9</u>	12.7	10.0	-	-	-
Video-LLaVA-7B (Lin et al., 2024)	95.1	20.3	17.8	-	-	-
PLLaVA-7B (Xu et al., 2024)	75.1	55.5	38.1	-	-	-
LLaVA-NeXT-Video-DPO-7B (Zhang et al., 2024a)	62.5	<u>60.9</u>	32.0	-	-	-
LLaVA-Video-7B-Qwen2 (Zhang et al., 2024b)	-	-	-	12.7	5.6	5.1
VideoLLaMA3-7B (Zhang et al., 2025a)	-	-	-	15.8	9.2	9.1
Qwen2.5-VL-7B-Instruct (Bai et al., 2025)	-	-	-	15.3	13.7	12.6
InternVL3.5-8B-Instruct (Wang et al., 2025b)	-	-	-	13.5	10.8	10.7
Qwen2.5-Omni-7B (base) (Qwen Team, 2025)	73.5	58.1	<u>44.8</u>	32.6	<u>17.8</u>	<u>19.1</u>
+ SODA (ours)	75.9	62.4	51.5	<u>32.1</u>	18.7	19.8

Table 3: General video understanding accuracy (%). VideoHalluciner reports Basic (factual), Hallucinated (counterfactual), and Overall accuracy where Overall requires both to be correct on the same video; “-” marks that the metrics are not available on the corresponding models. Per column, the best open-source result is in **blue bold** and the second-best is underlined. [†] quoted from official papers for all rows but ours.

takes a moving average of the student between rounds and *Joint update* reads the current student weights under a no-grad forward pass, while the audit, the voting, the loss, and the input budgets all stay as they are in the default frozen teacher. *Single-Stream* then narrows the audit itself, keeping only the questions seeded from the caption and dropping the ones seeded from the video. Finally, *SODA Round 1* and *Round 3* run the same recipe for fewer or more rounds, and the last row is the full recipe we use everywhere else in the paper.

Phase A produces the target, Phase B is what trains the model. The *Revised Caption Only* (c_r) row evaluates the revised caption c_r produced by Phase A on the base model, with no Phase B training applied. It already moves the base on Omni-Cloze visual and total, confirming that the audit-and-revise pipeline produces a meaningfully better caption than the model’s own draft. The full SODA recipe then gains a further, comparable amount on top, particularly on audio sub-scores and on coverage rate. The gap between *Revised Caption Only* (c_r) and full SODA is the share of the improvement that is *not* accessible by prompting harder at inference: Phase B’s per-token KL turns the one-shot textual lift into a property of the parameters, which the student carries to videos it has never seen revisions for and to inputs where the audit pipeline is unavailable. As the *#Input Tok.* column shows, Phase B also compresses the Phase A inference cost by $\sim 8\times$; Appendix Figure 3 plots the Pareto picture explicitly.

The input gap is the signal, a moving teacher doesn’t mean better. Because student and teacher share θ , when the teacher’s weights are read is a genuine design choice. We compare (i) *frozen teacher* (our default), with the checkpoint fixed at pre-Phase-B; (ii) *EMA update*, refreshed as an exponential moving average of the student between rounds; and (iii) *joint update*, where the teacher branch always reads the current student weights under a no-grad forward pass. EMA tracks the frozen teacher closely; joint update is consistently slightly worse, most visibly on Video-MME v2. The reason, we suggest, is structural: the teacher already differs from the student through access to c_s , $\mathcal{E}_{\text{rgb}}$, and a higher-fidelity audio-visual view, so stacking a moving-weight gap on top of an already informative input gap mostly introduces optimization noise without changing what is being taught. Freezing the teacher is both the simplest and the most stable choice.

Each stream owns one side, together they cover both. The *Single-Stream* ablation drops the video-grounded pivot stream, leaving the audit with only questions seeded from c_s . On VCapsBench-Omni this has the expected shape: IR *improves* because the student is only ever asked to defend claims it actually made, while CR *drops* because dimensions the caption never raised cannot be questioned at all. The dual-stream default trades a small amount of precision for a larger amount of recall, which is the right side of the curve on benchmarks where omission, not commission, is

Setting	Omni-Cloze				VCapsBench-Omni			Video-MME v2 (w/o sub)			#Input Tok.↓
	Vis↑	Aud↑	AV↑	Tot↑	AR↑	IR↓	CR↑	L1	L2	L3	
Qwen2.5-Omni-7B (Qwen Team, 2025)	18.3	14.1	21.9	16.6	44.5	29.7	60.7	32.6	17.8	19.1	830
Revised Caption Only (c_r)	27.3	13.0	22.3	21.8	45.8	26.6	62.4	-	-	-	6,936
w/o Extra Vid/Aud Info	26.5	20.0	24.4	23.9	46.8	29.1	67.8	31.3	17.2	19.1	830
w/o Audit	20.3	8.8	16.9	15.9	45.4	29.2	63.8	30.7	16.5	18.6	830
EMA update	26.7	22.2	25.0	<u>24.9</u>	<u>47.6</u>	30.1	<u>68.2</u>	31.3	<u>18.6</u>	19.0	830
Joint update	25.9	22.5	22.8	24.3	47.2	29.1	66.5	31.6	18.7	<u>19.6</u>	830
Single-Stream	25.0	<u>23.5</u>	24.4	24.6	46.2	<u>28.1</u>	65.6	30.2	17.8	18.5	830
SODA Round 1	26.7	19.2	22.3	23.6	<u>47.6</u>	29.4	67.8	<u>32.0</u>	17.4	19.0	830
SODA Round 3	22.4	24.9	20.9	23.0	45.8	29.9	65.3	30.5	18.7	19.8	830
SODA (Frozen Teacher+Dual Stream+Round2)	<u>27.1</u>	21.9	<u>24.8</u>	25.0	47.7	30.3	68.4	32.1	18.7	19.4	830

Table 4: Ablation studies; unless the row name says otherwise, every variant is trained for 2 rounds. The variants are defined in Section 4.4. The first row repeats the base Qwen2.5-Omni-7B numbers from Tables 2 and 3. *#Input Tok.*: average text tokens consumed at *inference* time per video to produce the evaluated caption (excluding video frame and audio tokens). Per column, the best ablation variant is in **blue bold** and the second-best is underlined (the reference row is excluded from the comparison).

the dominant failure mode. Caption-grounded for *check*, video-grounded for *recover*: neither subsumes the other.

Self-evolution helps, then plateaus. Treating the base model as Round 0 and refreshing Phase A on the round- r student each iteration, the procedure improves through Round 2 and then mildly regresses at Round 3 on Omni-Cloze (Appendix Figure 4), while the Video-MME v2 columns of Table 4 stay roughly flat. We attribute this pattern to a gradual saturation of the audit: as the student improves, we expect the pool of dimensions on which the audit can still extract a confident correction to shrink, so each additional refresh both adds less new signal and risks reinforcing the student’s residual biases against itself. We take Round 2 as the default operating point.

4.5 Generalization to Other Base Models

We also study the generalization ability of SODA by repeating the whole pipeline on a smaller model, Qwen2.5-Omni-3B, and on a different model family, MiniCPM-o-2.6.

Backbone	Benchmark	base	+SODA	Δ
Qwen2.5-Omni-3B	Omni-Cloze (Tot↑)	17.24	19.61	+2.37
	VCapsBench-Omni (AR↑)	37.6	42.51	+4.91
	VCapsBench-Omni (CR↑)	52.6	60.87	+8.27
	Video-MME v2 (Acc↑)	25.56	25.83	+0.27
MiniCPM-o-2.6	Omni-Cloze (Tot↑)	22.79	23.58	+0.79
	VCapsBench-Omni (AR↑)	30.6	33.81	+3.21
	VCapsBench-Omni (CR↑)	41.7	45.64	+3.94
	Video-MME v2 (Acc↑)	23.09	23.75	+0.66

Table 5: Generalization studies of SODA applied to different omni-modal backbones under the same training protocol. Video-MME v2 is reported here as overall accuracy, while Table 3 reports the benchmark’s per-level scores.

Table 5 shows that both additional backbones move in the same direction as the 7B result, so the gain does not come from one particular checkpoint. It is bigger on the Qwen2.5-Omni family

and smaller on MiniCPM-o-2.6, which fits how SODA gets its signal: a model that follows instructions less well writes a noisier Revision QA Bank, and a noisier bank gives a weaker revision target. Even then, coverage rate still goes up, so a weaker base model limits the method rather than breaking it. General video understanding again barely moves, which is the same picture we saw in Table 3.

5 Conclusion

In this paper, we start from two structural asymmetries that already live inside current VideoLLMs: captioning tends to be harder than answering questions sampled from a caption, and the same model reasons more reliably with a richer input budget. We then turn these into a self-distillation loop. The same model plays a high-budget teacher and a low-budget student; a dual-stream audit, filtered by self-consistency voting, builds a Revision QA Bank $\mathcal{E}_{\text{rqb}}$ that flags which caption dimensions need fixing; a token-level KL on the revised caption then transfers the teacher’s evidence into the student’s parameters without any human labels. Across three caption-fidelity benchmarks and video-understanding benchmarks, this single recipe improves caption fidelity, reduces hallucination, and does not erode general video QA, and it moves two further backbones in the same direction, so the gain does not rest on one checkpoint. Because the audit and the high-budget view are training-only, the distilled student keeps the base model’s inference cost. Two insights generalize beyond captioning: a “stronger teacher” can simply be the same model under a richer input budget, and which tokens carry the supervision matters as much as how much supervision is provided.

Limitations

We believe a systematic scaling-law study along both the model and data fronts is an important step in the future. Beyond scale, the self-supervised nature of SODA comes with a natural trade-off. We hope to scale up this approach in the future by injecting a small pool of ground-truth in a semi-supervised setup, or by injecting auxiliary meta-information beyond the raw audio-visual stream.

Acknowledgements

This research was supported by the Australian Government through the Australian Research Councils Discovery Projects funding scheme (project numbers DP260100218 and DP260101891). Dr Qiuhong Ke is the recipient of an Australian Research Council Discovery Early Career Researcher Award (project number DE250100030) funded by the Australian Government.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.
- Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. 2024a. ShareGPT4Video: Improving video understanding and generation with better captions. In *NeurIPS Datasets and Benchmarks Track*.
- Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. 2024b. Panda-70M: Captioning 70M videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-LLMs. *arXiv preprint arXiv:2406.07476*.
- Yanqi Dai, Yong Wang, Zebin You, Dong Jing, Xiangxiang Chu, and Zhiwu Lu. 2026. Adaptive Task Balancing for Visual Instruction Tuning via Inter-Task Contribution and Intra-Task Difficulty. In *Proceedings of the ACM Web Conference (WWW)*.
- Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. 2026. Video-MME-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*.
- Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *ICML*.
- Google DeepMind. 2025a. Gemini 2.0 Flash. *Google blog post*.
- Google DeepMind. 2025b. Gemini 2.5 Flash. *Google blog post*.
- Google DeepMind. 2025c. Gemini 2.5 Pro. *Google blog post*.
- Google DeepMind. 2025d. Gemma 3n. *Google blog post*.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *ICLR*.
- Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *CVPR*.
- Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.
- Martin Klissarov, Jonathan Cook, Diego Antognini, Hao Sun, Jingling Li, Natasha Jaques, Claudiu Musat, and Edward Grefenstette. 2026. Improving interactive in-context learning from natural language feedback. *arXiv preprint arXiv:2602.16066*.
- Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. In *ICML*.

- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *CVPR*.
- Chaoyu Li, Eun Woo Im, and Pooyan Fazli. 2025a. VidHalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In *CVPR*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *EMNLP*.
- Zongxia Li, Xiyang Wu, Guangyao Shi, Yubin Qin, Hongyang Du, Fuxiao Liu, Tianyi Zhou, Dinesh Manocha, and Jordan Lee Boyd-Graber. 2025b. VideoHallu: Evaluating and mitigating multi-modal hallucinations on synthetic video understanding. In *NeurIPS*.
- Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-LLaVA: Learning united visual representation by alignment before projection. In *EMNLP*.
- Ziyang Ma, Ruiyang Xu, Zhenghao Xing, Yunfei Chu, Yuxuan Wang, Jinzheng He, Jin Xu, Pheng-Ann Heng, Kai Yu, Junyang Lin, Eng Siong Chng, and Xie Chen. 2026. Omni-captioner: Data pipeline, models, and benchmark for omni detailed perception. In *ICLR*.
- OpenAI. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*.
- OpenBMB. 2024. MiniCPM-o 2.6: A GPT-4o level MLLM for vision, speech, and multimodal live streaming on your phone. *Preprint*.
- Qwen Team. 2025. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.
- Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. 2026. CRISP: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, and 22 others. 2024. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research (TMLR)*.
- Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. video-SALMONN: Speech-enhanced audio-visual large language models. In *ICML*.
- Kim Sung-Bin, Oh Hyun-Bin, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. 2025. AVHBench: A cross-modal hallucination benchmark for audio-visual large language models. In *ICLR*.
- Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. 2025. video-SALMONN 2: Captioning-Enhanced Audio-Visual Large Language Models. *arXiv preprint arXiv:2506.15220*.
- Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*.
- Peiyao Wang, Haotian Xu, Noranart Vesdapunt, Rui Hou, Jingyi Zhang, Haibin Ling, Oleksandr Obiednikov, Ning Zhou, and Kah Kuen Fu. 2026. Reinforcing structured chain-of-thought for video understanding. *arXiv preprint arXiv:2603.25942*.
- Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2025a. VisualPRM: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.
- Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. 2024. VideoHalluciner: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*.

- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *ACL*.
- Peiran Wu, Yunze Liu, Zhengdong Zhu, Enmin Zhou, and Junxiao Shen. 2025. UGC-VideoCaptioner: An omni UGC video detail caption model and new benchmarks. *arXiv preprint arXiv:2507.11336*.
- Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves ImageNet classification. In *CVPR*.
- Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. PLLaVA: Parameter-free LLaVA extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.
- Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*.
- Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Haitao Zheng, Maosong Sun, and Tat-Seng Chua. 2024. RLHF-V: Towards trustworthy MLLMs via behavior alignment from fine-grained correctional human feedback. In *CVPR*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *NeurIPS*.
- Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. 2025a. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.
- Jiacheng Zhang, Yang Jiao, Shaoxiang Chen, Na Zhao, Zhiyu Tan, Hao Li, Xingjun Ma, and Jingjing Chen. 2025b. EventHallusion: Diagnosing event hallucinations in video LLMs. *arXiv preprint arXiv:2409.16597*.
- Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *ICCV*.
- Shi-Xue Zhang, Hongfa Wang, DuoJun Huang, Xin Li, Xiaobin Zhu, and Xu-Cheng Yin. 2025c. VCaps-Bench: A large-scale fine-grained benchmark for video caption quality evaluation. *arXiv preprint arXiv:2505.23484*.
- Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024a. LLaVA-NeXT: A strong zero-shot video understanding model. *Blog post*.
- Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024b. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.
- Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. 2024. Beyond hallucinations: Enhancing LVLMS through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*.
- Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. 2024a. Aligning modalities in vision large language models via preference fine-tuning. In *ICLR*.
- Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024b. Analyzing and mitigating object hallucination in large vision-language models. In *ICLR*.
- Orr Zohar, Xiaohan Wang, Yonatan Bitton, Idan Szepes, and Serena Yeung-Levy. 2025. Video-STaR: Self-training enables video instruction tuning with any supervision. In *ICLR*.

A Appendix

A.1 Experimental Setup Details

This subsection collects the hyperparameters and implementation details deferred from Section 4.1.

Information lifts. The student view V_{lo} uses $F_s \leq 100$ frames sampled at 1 fps from a per-frame pixel budget $R_s = 151,200$ and a $T_s = 120$ s audio window at 16 kHz mono; the teacher view (V_{hi}, A_{hi}) uses $F_t \leq 100$ frames at 2 fps from a per-frame pixel budget $R_t = 453,600$ ($3 \times$ the student) and a $T_t = 240$ s audio window ($2 \times$ the student), so the teacher consistently observes a richer audio-visual budget. Phase A captions are sampled with temperature 0.7 inside a length window of $[80, 4096]$ tokens, and the QA generator emits exactly 28 questions per video. Self-Voting uses $K = 5$ samples per question under temperature $T = 0.7$ and nucleus $p = 0.9$, with agreement thresholds $\tau = 0.6$ for Stream-1 retention and $\tau_{piv} = 0.6$ for Stream-2 pivot triggering, capped at $D_{max} = 8$ pivot dimensions per video. Phase A is refreshed every R optimization steps.

Implementation details. We instantiate SODA on Qwen2.5-Omni-7B. Phase A runs in inference mode with data-parallel sharding across the available accelerators; Phase B uses standard distributed data-parallel training. Phase B trains LoRA adapters of rank 64 with $\alpha = 128$ on the $\{q, k, v, o\}_{proj}$ matrices of the language backbone,

keeping the visual and audio encoders frozen. Optimization uses AdamW with a constant learning rate 5×10^{-5} and 5% linear warm-up, per-device batch size 2 with gradient accumulation 1, maximum sequence length 32,768 tokens, and a maximum completion length of 4,096 tokens. The loss is the KL form in Eq. 6 with softmax temperature 1.0; we optionally replace it with Jensen-Shannon (mixture coefficient $\beta = 0.7$). Training runs for 2 on-policy rounds; each round consists of one Phase A cache regeneration followed by one Phase B epoch over a 2,000-video subset sampled from Panda-70M, and successive rounds warm-start from the previous round’s merged adapter.

A.2 Asking the Question Directly: a QA-vs-Caption Probe on the Base Model

The dual-stream audit in Phase A is built on the claim that, for the same omni-modal model, *answering a yes/no question grounded in audio-visual evidence is a more accurate task than writing a free-form caption and reading the answer back out of the caption* (Section 3.1, “task asymmetry between captioning and question answering”). This subsection reports the simplest possible direct check of that claim on the base model SODA is built on.

Setup. We hold the model fixed at the base Qwen2.5-Omni-7B and run two strictly parallel pipelines on the same 100-video subset of VCapsBench-Omni (seed-42 random draw), using the yes/no QAs in the intersection across visual, audio, and AV-joint tracks.

- **Caption→judge.** The same protocol as the main-table evaluation (Section 4.2). The model emits one omni caption per video; Gemini-2.5-Pro then reads each GT QA off the caption text alone and returns {YES, NO, UNANSWERABLE}.
- **Direct-QA.** For each GT QA, the same model is shown the video and audio with a one-line yes/no/unsure instruction and is asked to commit to a verdict directly. No external judge is involved; the model’s own answer is graded against GT.

The two pathways score with the same AR / IR / CR formula on the same (video, question) pairs, so the comparison is apples-to-apples. The judge in the caption pathway is strictly stronger than the model, which biases the comparison *against* our hypothesis.

Results. Table 1 in the main paper reports the overall numbers, and Table 6 together with Figure 1 give the per-modality view. The direct-QA pathway is more accurate by a large margin on every track, gaining +33.8 AR overall and improving IR and CR as well. The per-modality breakdown is the predicted shape: the gap is smallest on visual, where captions tend to enumerate scene content anyway, and widest on audio and AV-joint, where free-form omni captions routinely omit the dimension entirely and the judge has no choice but to mark the QA UNANSWERABLE. CR makes this concrete: on AV-joint the caption pathway can only attempt about two out of five QAs, while direct-QA commits to nearly all of them. Figure 1 confirms this at finer resolution: all but one of the VCapsBench-Omni dimensions that move between the two pipelines move upward, and the largest jumps concentrate on the audio (red) and AV-joint (purple) clouds in the upper-left.

Per-QA agreement. The same conclusion sharpens at the per-QA level. On the intersected QAs, the direct-QA pathway recovers several times more QAs than it loses. The overwhelming majority of those direct-QA-only wins are cases where the caption pathway returned UNANSWERABLE (the caption was silent on that dimension) and the model nonetheless committed to the correct yes/no when asked directly. These are exactly the items that the Video QA Stream in Phase A is designed to surface and that the caption-only audit would miss by construction.

Why this motivates dual-stream auditing. The QA-vs-caption gap above is what the audit converts into supervision. The Caption QA Stream (Section 3.2) targets the cases where the caption *does* commit to a claim but the claim is wrong, which is the IR row of Table 1 and corresponds to the *check* job. The Video QA Stream targets the large share of probes that the caption stays silent on ($1 - \text{CR}$, overall column) while the model can answer them directly, which is the *recover* job. Both classes are large, both are concentrated on the audio and AV-joint sides, and both vanish from any supervision signal that lives on the caption text alone. Folding both into c_r and then distilling c_r back into the student in Phase B is the mechanism by which SODA turns “the model knows more than its caption admits” into a parameter-level gain.

	AR $\uparrow$	IR $\downarrow$	CR $\uparrow$
<i>Visual</i> ($n=1,716$)			
caption $\rightarrow$ judge	58.1	17.1	70.1
direct-QA	83.2	12.8	95.4
Δ	+25.1	-4.3	+25.3
<i>Audio</i> ($n=1,059$)			
caption $\rightarrow$ judge	31.0	42.9	54.2
direct-QA	68.8	16.6	82.5
Δ	+37.9	-26.3	+28.3
<i>AV-joint</i> ($n=652$)			
caption $\rightarrow$ judge	25.7	32.8	38.2
direct-QA	75.5	14.7	88.5
Δ	+49.8	-18.1	+50.3

Table 6: Per-modality Caption $\rightarrow$ judge vs. Direct-QA on the same 100-video probe as Table 1. The judge for the caption pathway is Gemini-2.5-Pro. Direct-QA is graded against GT without any judge. Δ rows are direct-QA minus caption $\rightarrow$ judge.

A.3 Supplementary Figures for Ablations

This subsection collects two visualizations of the ablation results in Table 4, both referenced from Section 4.4.

A.4 Phase A Audit Dimension Taxonomy ($D=29$)

The dual-stream audit in Phase A is organized around a fixed taxonomy of $D=29$ audio-visual dimensions, designed to give the QA generator a stable, enumerable set of facets to check against. The taxonomy is drawn from the evaluation dimensions covered by VCapsBench (Zhang et al., 2025c) on the visual and camera/aesthetic side, together with an audio and audio-visual taxonomy we define below, but is intentionally coarser than either grid: a tight set of 29 dimensions keeps both QA streams short enough to fit into a single audit pass while still spanning the perceptual aspects on which Video LLMs tend to drop content. We list the dimensions in the exact snake-case naming used by our QA-generator system prompt so that this appendix and the audit prompts can be cross-referenced one-to-one. The 29 dimensions are grouped into 5 macro categories.

Visual content (15). (1) `subject_existence`: which entities, people, or objects appear, their counts, and when they enter or leave; (2) `subject_attr`: identity, age, gender, profession, body features, material, shape, size; (3) `subject_relation`: comparative, family, hierarchical, or identity relations between subjects; (4) `action_interaction`: fine-grained actions and agent-to-object or agent-to-agent interactions; (5) `state_change`: how subjects or objects

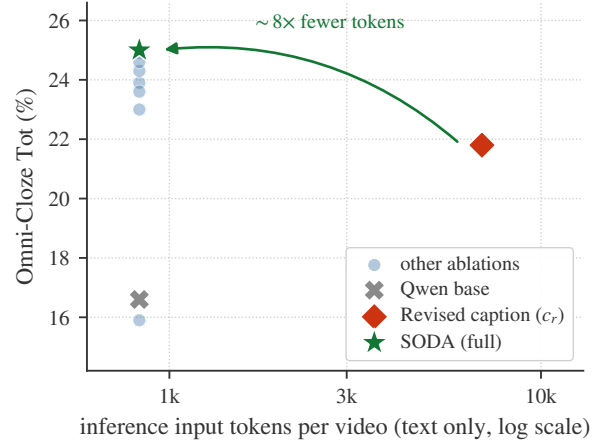


Figure 3: Inference cost vs. caption-fidelity quality across the ablation variants of Table 4. The x -axis is per-video text input tokens at inference time (log scale; video frame and audio tokens excluded); the y -axis is Omni-Cloze Tot. accuracy. *Revised Caption Only* (c_r) sits at $\sim 6.9k$ tokens because every inference must re-run the full Phase A pipeline, while all Phase B-trained variants run the student on the standard caption prompt alone (830 tokens) and the full SODA recipe dominates on both axes simultaneously.

change state over time; (6) `spatial_layout`: 2D positions, left/right/above/below, direction, orientation; (7) `scene_setting`: place, environment, indoor/outdoor, background cues; (8) `emotion_atmosphere`: emotions of people, overall mood and atmosphere; (9) `world_knowledge`: culturally or historically specific items, famous people, brands, proper nouns; (10) `lighting_color`: lighting style and overall color palette/tone; (11) `fine_detail`: small peripheral details, textures, decorations, logos, patterns; (12) `clothing_accessories`: clothing items and accessories of each visible person; (13) `color_specific`: precise color of each named object (not the overall palette); (14) `counting_precise`: exact integer counts for distinct subject types; (15) `scene_segments`: how many scenes/shots the video contains and what each one depicts.

Camera and aesthetic (5). (16) `camera_movement`: camera motion type (pan, tilt, zoom, tracking, dolly, handheld, static, orbit) and how it changes; (17) `shot_type`: shot scale and framing (close-up, medium, wide, establishing, POV, aerial, over-the-shoulder); (18) `blur_dof`: depth of field (shallow bokeh vs. deep focus), focus pull, and motion blur; (19) `color_grading`: post-processing color treatment (warm, cool, vintage, vivid, muted, high-

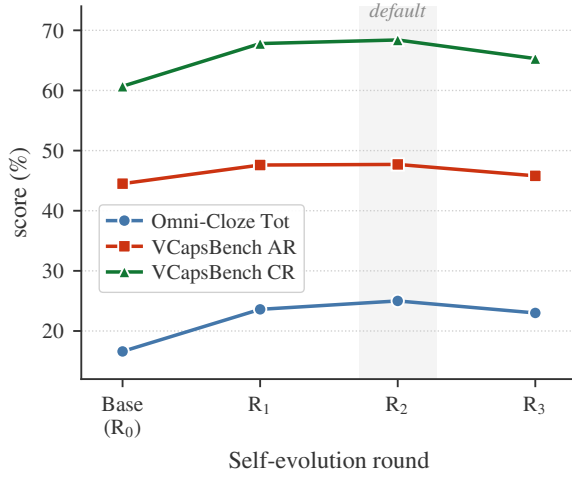


Figure 4: Self-evolution trajectory across rounds. Round 0 is the base Qwen2.5-Omni-7B. We train two on-policy rounds as the default and additionally report a third round. All three caption-fidelity metrics improve through Round 2 and then mildly regress at Round 3, consistent with a gradual saturation of the audit as the student improves.

contrast, desaturated, tinted); (20) `visual_style`: production or artistic style (cinematic, documentary, first-person, animation, timelapse, UGC vlog, AI-generated).

On-screen text (2). (21) `native_ocr`: real-world text inside the scene (signs, handwriting, labels, packaging); (22) `overlay_text`: superimposed subtitles, captions, or graphic overlays, including their content, position, and duration.

Audio (5). (23) `speech_content`: what is spoken, language, dialect, speaker identity, and transcribable phrases; (24) `voice_attribute`: voice timbre, pitch, emotion, delivery style (whisper, shout, sung, narration); (25) `audio_event`: non-speech sounds (music, instrument, animal, object, mechanical, SFX) and their sources; (26) `sound_scene`: acoustic environment inferred from audio (reverberant indoor, open outdoor, underwater, ...); (27) `av_sync`: audio-visual correspondence, on-screen vs. off-screen sound, narration timing.

Cross-modal (2). (28) `anti_hallucination`: probe whether described elements are actually present, and which commonly-assumed elements are NOT; (29) `content_type`: video category (educational, film/TV, social-media UGC, commercial, documentary, ad).

Each dimension comes with a one-line “what to ask” specification in the QA-generator system prompt, which the Caption QA Stream and the

Video QA Stream both consume to generate q_i . For the Caption QA Stream, the generator is additionally allowed to return `SKIP_NOT_IN_CAPTION` on dimensions that the initial caption c_s does not address; the dropped dimensions go to S_{skip} and become the seed of the Video QA Stream pivot (Section 3.2).

Audit pipeline statistics. Table 7 summarizes what Phase A actually produces when the audit is run over the 2,000-video training set in our default Round 1 frozen-teacher configuration. We measure two complementary layers. The *caption-revision* numbers (top block) are computed directly from the audit JSONL that Phase A writes: how often the audit fires and how much the caption is actually rewritten. The *caption-vs-audit* numbers (bottom block) ask a different question: for every probe $(c_s, q_i, \text{winner}_i)$ that survives Self-Voting and enters $\mathcal{E}_{\text{rqb}}$, is the student caption c_s consistent with the voted audit answer winner_i ? We answer this by sending each triple to a strong LLM judge (Gemini-2.5-Pro), which returns one of `{CONSISTENT, INCONSISTENT, ABSENT, UNCERTAIN}`; the four fields below summarize the resulting distribution.

Field definitions. (i) *Videos with $c_r \neq c_s$* : fraction of videos for which the audit actually triggers a rewrite. (ii) *Mean character-level diff*: $|c_s - c_r|$ as a fraction of $|c_s|$, computed by Python’s `difflib` ratio. (iii) *Caption-present rate*: judge marked the caption as making a claim on that dimension (`CONSISTENT` or `INCONSISTENT`). (iv) *Caption-correct | present*: among those, fraction marked `CONSISTENT`; this is the precision of what the caption already says. (v) *Caption-commission rate*: fraction of *all* probes marked `INCONSISTENT`; the caption made a claim but the audit disagreed. (vi) *Caption-absent rate*: fraction marked `ABSENT`; the caption was silent on that dimension. By construction, `present + absent + uncertain` ≈ 1 .

What the numbers say. Two patterns are worth flagging. (i) The audit triggers an actual rewrite on a clear majority of videos, while keeping the median edit small, consistent with the design intent of a surgical revision step rather than a full rewrite. (ii) Where the audit really earns its keep is precision and recall jointly: the judge marks a modest share of probes as *inconsistent* with the voted audit, and a much larger share as *absent*, with the absent share far higher on audio than on visual. This confirms the design motiva-

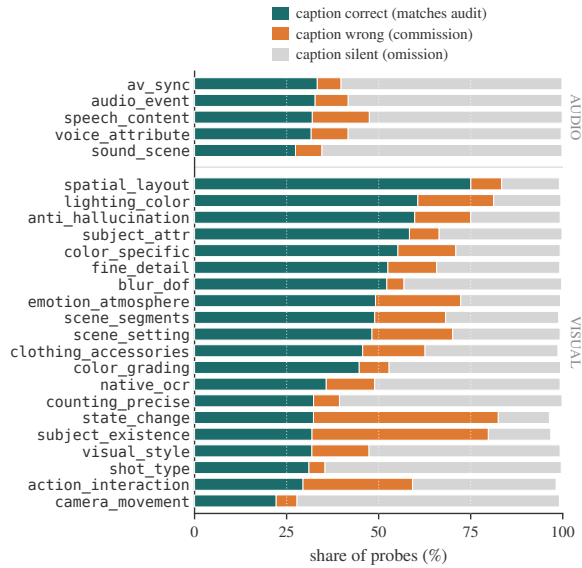


Figure 5: Per-dimension caption-vs-audit decomposition from Table 7 (Gemini-2.5-Pro judge). For each of the 25 active dimensions, the stacked bar shows the share of probes where the initial caption c_s is correct (teal, matches the voted audit answer), wrong (orange, commission), or silent (gray, omission). Rows are grouped by modality (audio block on top) and sorted within each group by correct-rate.

tion for the Video QA Stream: the caption leaks roughly a third of visual dimensions and well over half of audio ones, and a caption-only audit would miss those by construction. Figure 5 shows the per-dimension decomposition. The audio block at the top is dominated by absent (gray) bars, confirming that omission, not commission, is the bottleneck on the audio side. Within the visual block, dimensions that require describing high-level facts the model is well-trained on (spatial_layout, lighting_color) are mostly correct, whereas dimensions that demand commitment to fine factual content (state_change, subject_existence, action_interaction) accumulate the largest commission stripes, and camera-related dimensions (camera_movement, shot_type) are silent on a clear majority of clips. The figure makes the audit’s design tractable: a single training pass extracts both the unwritten audio content and the over-confident state-change claims, two failure modes that the caption alone never surfaces.

A.5 VCapsBench-Omni: Data Construction Details

This subsection details the construction of *VCapsBench-Omni*, the omni-modal extension of *VCapsBench* (Zhang et al., 2025c) that we use to

Quantity	Value
<i>Caption-level Revision</i>	
Videos with $c_r \neq c_s$	61.0%
Mean character-level diff between c_s and c_r	3.2%
<i>Caption-Audit Consistency</i>	
Caption-present rate, overall	57.2%
visual / audio	63.2% / 41.2%
Caption-correct present, overall	71.2%
visual / audio	70.1% / 75.7%
Caption-commission rate, overall	16.5%
visual / audio	18.9% / 10.0%
Caption-absent rate, overall	41.9%
visual / audio	35.7% / 58.6%

Table 7: Phase A audit statistics on the 2,000-video training set, Round 1 (frozen-teacher default). The top block is computed directly from the audit JSONL; the bottom block reports an LLM-as-judge (Gemini-2.5-Pro) assessment over the surviving probes of whether c_s is consistent with the audit’s voted answer.

evaluate audio and audio-visual joint caption fidelity in Section 4. The construction protocol is intentionally a faithful re-implementation of the original *VCapsBench* data-construction pipeline (Figure 4 of Zhang et al., 2025c), with three deliberate changes:

1. the visual yes/no QA set is reused *verbatim* from *VCapsBench* (no re-generation, no re-curation), so that the visual track of *VCapsBench-Omni* is comparable, video-by-video and QA-by-QA, with the original benchmark;
2. we add two new tracks on top of the visual one, namely an *audio* track (4 macro / 13 sub categories) and an *audio-visual joint* track (1 macro / 5 sub categories), and use the same two-pipeline-plus-correction protocol to construct them;
3. all VLM and LLM calls in the new tracks are issued to a single annotator model, Gemini 2.5 Pro, which acts both as the multimodal annotator (Pipelines 1 and 2) and as the textual curator (Data Correction Pipeline).

The remainder of this appendix describes the taxonomy, the three pipelines, and the final merging step. All prompts and QA strings are in English.

A.5.1 Taxonomies

The visual taxonomy is identical to *VCapsBench*: 4 macro categories and 21 sub categories covering action, count, text, color, position, relative position, entity, entity shape, entity size, background, relation, lighting, style, shot type, atmosphere, camera movement, color palette, color grading,

blur, emotion, and proper noun. We do not modify it.

For audio, we use the following 4 macro categories with 13 sub categories:

- **Human_Voice_Speech:** Speaking_State, Non_Verbal_Vocalization;
- **Physical_SFX:** Action_SFX, Rigid_Body_Collision, Fluid_Sound, Mechanical_Sound, Percussion_SFX;
- **Ambience:** Natural_Ambience, Social_Ambience, Silence;
- **Background_Music:** Emotional_Tone, Music_Genre, Rhythm_Tempo.

For the audio-visual joint track, we restrict ourselves to a single macro category (**AV_Joint**) with 5 sub categories that all require evidence from *both* modalities to be answered: *Sound_Source_Grounding*, *Lip_Sync*, *Action_SFX_Alignment*, *AV_Emotion_Consistency*, and *Off_Screen_Sound*. We deliberately exclude acoustic-temporal questions (precise sound onset / offset, event count) because current models, including the strongest closed-source ones, are unreliable on those.

A.5.2 Pipeline 1: Caption-then-QA

Pipeline 1 mirrors the upper branch of Figure 4 of [Zhang et al. \(2025c\)](#) and is split into two stages.

Stage 1.1: detailed audio + AV-binding caption.

For every video, Gemini 2.5 Pro is asked to watch the clip and emit one detailed English caption that thoroughly covers the audio content and the audio-visual binding. Visual-only details are deliberately kept brief, since the visual side is already covered by VCapsBench’s native QA. The model is instructed to (i) enumerate every salient sound (human voice, physical SFX, ambience, music) and its perceptual attributes (pace, tone, emotion, genre, tempo); (ii) bind salient sounds to their visible emitters whenever possible (source localization, lip sync, action-to-SFX alignment, AV emotion consistency); and (iii) mark off-screen sounds explicitly. The caption is constrained to 80 to 200 plain-prose English words, with no hedging language (“maybe”, “probably”), and is returned as a STRICT-JSON object with a single `caption` field.

Stage 1.2: LLM-mined yes/no QAs from the caption. The captions from Stage 1.1 are then fed, video-by-video, to Gemini 2.5 Pro *in text-only mode*. The model is asked to mine two kinds of yes/no QA-pairs that must be derivable from

the caption alone: `audio_qa` (answerable from the audio content described in the caption) and `av_joint_qa` (require both an audio and a visual cue described in the caption). The prompt enforces several hard constraints:

- Each question is a binary yes/no question and answer is exactly “yes” or “no”; the LLM is forbidden from inventing facts not present in the caption.
- Each audible event is decomposed into multiple atomic QAs covering existence / type / attribute / source binding.
- For every salient verifiable absence in the caption (e.g. “no music”), one or two negative (answer: no) audio QAs are produced.
- Every `av_joint_qa` question must *simultaneously* reference a visual cue and an audio cue; the only valid AV-joint types are the five listed in Section A.5.1.
- Per-video budgets are 5 to 10 audio QAs and 4 to 8 AV-joint QAs; near-silent clips emit zero QAs.
- Categories are picked strictly from the audio and AV-joint taxonomies, using the exact English tags.

The LLM emits a STRICT-JSON object with two fields, `audio_qa` and `av_joint_qa`, each entry carrying (`concept`, `question`, `answer`, `category_1`, `category_2`).

A.5.3 Pipeline 2: Direct QA from the Video

Pipeline 2 mirrors the lower branch of Figure 4 of [Zhang et al. \(2025c\)](#): instead of going through an intermediate caption, Gemini 2.5 Pro watches the video directly and emits yes/no QAs grounded in what it can see and hear. The taxonomy and per-category budgets are identical to those in Pipeline 1; the same atomic-decomposition and salient-absence rules apply, and the same hard constraint (every AV-joint question must reference a visual *and* an audio cue) is enforced. The resulting QAs play the same role as the corresponding lower branch in the original VCapsBench Figure 4, providing an independent source that is later cross-checked against Pipeline 1.

A.5.4 Data Correction Pipeline

The QA-pool produced by merging Pipelines 1 and 2 is cleaned in two stages, exactly mirroring the data-correction pipeline in Figure 4 of [Zhang et al. \(2025c\)](#).

Stage A: rule-based pre-filter. For every QA in the merged pool we record a `sources` tag indicating which of $\{p_1, p_2\}$ produced it, and a `per_source_answers` list with the yes/no answer each pipeline gave. The rule layer is deterministic and cheap:

1. *Conflict drop*: any QA whose `per_source_answers` contains both “yes” and “no” is dropped (always on);
2. *No-concept drop*: any QA with an empty `concept` field is dropped (on by default; suggests low-quality generation);
3. *Singleton drop* (optional): any QA produced by only one of the two pipelines is dropped, mirroring the “appears only once” filter described in the original VCapsBench paper.

Stage B: LLM curator. For every (video, category_2) bucket, the rule-filtered QAs are passed to Gemini 2.5 Pro acting as a senior data curator. The curator is asked to:

1. *Merge near-duplicates*. If two questions ask the same underlying audio fact in different wording, keep exactly one (preferring the clearer / more specific phrasing).
2. *Drop low-quality items*. Drop QAs whose questions are vague, ambiguous, unanswerable from audio/AV evidence, or near-tautological.
3. *Preserve ground-truth answers as given*. Yes/no flips are explicitly forbidden.
4. *Respect the category*. QAs cannot be moved to a different category.

Bucket sizes are small by construction (one video $\times$ one sub category), so each curator call is cheap. On curator failure we fall back to the rule-filtered output for that bucket.

A.5.5 Final Merge and Format

Three sources are merged to produce the final dataset: (i) the video list from a balanced 1k subset of VCapsBench, (ii) the visual yes/no QAs taken *verbatim* from VCapsBench_100KQA.jsonl, and (iii) the audio and AV-joint QAs produced by the data-correction pipeline. Each output line carries the visual QAs in the original VCapsBench triple-string format, the structured audio QAs, the structured AV-joint QAs, and “flat” mirrors that re-encode the audio and AV-joint QAs into the same triple-string format so that any VCapsBench-style evaluator can also consume them directly. The audio caption from Pipeline 1 is retained as a refer-

ence field but is not part of the evaluation set.

A.5.6 Per-model Evaluation Protocol

Per-model evaluation also follows the standard VCapsBench protocol. The model under test first emits one English omni caption per video. An LLM judge (also Gemini 2.5 Pro in our setup) then verifies each yes/no QA *solely from the caption text*; the eval scripts never re-watch the video. Each verdict is one of {Yes, No, Unanswerable}, mapped to the three standard rates: accuracy rate (AR $\uparrow$, fraction of QAs answered correctly), inconsistency rate (IR $\downarrow$, fraction answered incorrectly), and coverage rate (CR $\uparrow$ = 1 – uncertain rate, fraction the caption contains enough information to answer at all). The judge prompts are slightly track-specific: the visual judge is the standard “answer yes / no / unanswerable based only on caption” template; the audio judge prioritises the core audio event referenced by the question and ignores non-audio frills; the AV-joint judge is stricter and requires the caption to explicitly mention the visual fact, the audio fact, and the binding between them, otherwise the verdict is Unanswerable. The three rates are reported per track (visual / audio / AV-joint) and as a QA-count-weighted overall score.

A.6 Prompt Templates for Phase A and Phase B

This subsection makes the two prompt templates used throughout Section 3 explicit. The same two strings are reused across phases: the captioning prompt `promptc` conditions the student in Phase A draft generation (Eq. 1) and the student path in Phase B (Eq. 4); the revision prompt `promptr(cs, Ergb)` conditions the teacher in Phase A revision (Eq. 3) and the teacher path in Phase B (Eq. 5). Both are fixed strings; only `cs` and `Ergb` are spliced in at runtime. We show the bare-string content below; the backbone’s standard chat template wraps each block as a single user message.

Captioning prompt `promptc`. Deliberately minimal so that the student is forced to scan the audio-visual input without any text-side hint about which dimensions to cover.

You are given a short video with an audio track. Write one detailed English caption that describes both what you see and what you hear in the video, as a single paragraph of 80–200 words. Cover salient subjects, actions, on-screen text, camera work, background music, speech content, ambient sounds, and any

audio-visual binding (who is speaking, what makes a given sound). State only what the video supports. Do not use hedging language such as “maybe” or “possibly”.

Revision prompt $\text{prompt}_r(c_s, \mathcal{E}_{\text{rgb}})$. Splices the student draft c_s and the verified bank $\mathcal{E}_{\text{rgb}}$ (rendered as a numbered list of $(q_i, \hat{a}_i)$ pairs) into a fixed shell of revision principles. Angle-bracketed tokens mark per-example splice points; $\langle c_s \rangle$ is replaced by the verbatim text of the initial caption and each $\langle q_i \rangle / \langle a_i \rangle$ by the i -th surviving QA pair from $\mathcal{E}_{\text{rgb}}$.

```
You are revising a video caption.
[INITIAL CAPTION]
<c_s>
[VERIFICATION Q&A]
The following questions have been
verified against the audio-visual
content of the video. Each answer is the
verified ground truth and takes FINAL
authority over the initial caption.
1. Q: <q_1> → A: <a_1>
2. Q: <q_2> → A: <a_2>
...
N. Q: <q_N> → A: <a_N>
[INSTRUCTIONS]
- Rewrite the initial caption into
a single revised caption that is
consistent with every verified answer
above.
- Keep the spans that the verification
confirms; rewrite only the spans
that the verification contradicts or
under-specifies.
- For any audio question whose answer is
“none”, remove the corresponding audio
claim from the caption (do not assert
that sound).
- Do not introduce facts that are neither
in the initial caption nor in the
verification list.
- Do not use hedging language such as
“maybe” or “possibly”.
- Return one English caption of 80-200
words as a single paragraph.
```

How Phase B consumes these prompts. At Phase B training time, both prompts are rendered into the textual context that prefixes the same teacher-forced caption c_r . For each cached tuple $(c_s, \mathcal{E}_{\text{rgb}}, c_r)$, the student path encodes the low-budget view V_{lo} and prefixes prompt_c ; the teacher path encodes the high-budget view $(V_{\text{hi}}, A_{\text{hi}})$ and prefixes $\text{prompt}_r(c_s, \mathcal{E}_{\text{rgb}})$ under `torch.no_grad`. Both paths then read off logits over the same c_r token positions, and Eq. 6 averages the per-token KL between them over $|c_r|$. The two paths differ in both the textual prefix and the input budget, so the loss carries the text-side lift induced by $(c_s, \mathcal{E}_{\text{rgb}})$

together with the multimodal lift; the splice points in the revision prompt are re-evaluated only when Phase A is refreshed (every R optimization steps), and are otherwise read straight from the cache.

A.7 Phase A Walkthrough: A Real Example

To make the dual-stream audit and the revision step concrete, we walk through one real training example produced by Phase A, copied verbatim from the cached tuple $(c_s, \mathcal{E}_{\text{rgb}}, c_r)$. The example uses an in-the-wild silent video showing a finger pointing at two leaves on the soil. We display three items in turn: the student draft c_s , the QA pairs that survived Self-Voting (i.e. the contents of $\mathcal{E}_{\text{rgb}}$), and the revised caption c_r that the teacher writes conditioned on c_s and $\mathcal{E}_{\text{rgb}}$.

Student caption c_s (low-budget view).

The video opens with a close-up shot of two green leaves resting on a patch of soil. The soil appears to contain some dried plant matter and small pieces of wood. A few green plants are also visible in the background. A finger, likely belonging to the person holding the camera, points at the left leaf first and then the right leaf. The camera then zooms in for a closer look at the leaves, showcasing their texture and the veins running through them. The finger continues to point at both leaves, emphasizing specific areas. The camera remains focused on the leaves throughout, with the background remaining constant. There are no other people or objects in the frame. The video seems to be a detailed examination of the leaves, possibly for educational or observational purposes. The lighting is natural, suggesting an outdoor setting. There is no audio present in the video.

Revision QA Bank $\mathcal{E}_{\text{rgb}}$. We list each QA pair in $\mathcal{E}_{\text{rgb}}$ in the form “ $q \rightarrow \hat{a}(\rho)$ ”. Each $\hat{a}$ is the majority answer over $K=5$ teacher samples on the high-budget view, and ρ is the corresponding agreement rate; only pairs with $\rho \geq \tau$ and $\hat{a} \neq$ “uncertain” are retained.

- **Q1** (visual): What texture do the leaves have? → smooth ($\rho=0.67$).
- **Q2** (visual): What type of environment is the video shot in? → outdoor ($\rho=1.00$).
- **Q3** (visual): What is the color of the soil? → brown ($\rho=0.67$).
- **Q4** (blur_dof): What is the depth of field in the video? → shallow ($\rho=0.67$).
- **Q5** (audio): Is there any background music audible in the video? → none ($\rho=0.67$).
- **Q6** (audio): Is there any spoken dialogue in the video? → none ($\rho=1.00$).
- **Q7** (audio): Are there any non-music sound effects in the video? → none ($\rho=1.00$).
- **Q8** (audio): Is there any narration in the video? → none ($\rho=1.00$).

Two patterns are worth noting. First, Q1 to Q4 are visual-content checks that fill in fine-grained

attributes the student left implicit (*smooth* texture, *brown* soil, *shallow* depth of field), produced by the Caption QA Stream from c_s . Second, Q5 to Q8 are audio negatives with $\hat{a}=\text{none}$; these are the kind of dimensions c_s tends to omit and that the Video QA Stream is designed to recover. They survive Self-Voting at $\rho \geq 0.67$ because the teacher consistently reports silence across all K samples on the high-budget view.

Teacher prompt for revision. The teacher receives the student draft c_s followed by $\mathcal{E}_{\text{rqb}}$ as a numbered “Verification Q&A” list, plus a fixed set of revision principles (greedy decoding, single pass, ground-truth answers have FINAL authority over c_s , hedging words removed, audio claims with $\hat{a}_i=\text{none}$ must be removed). The full template is the revision prompt $\text{prompt}_r(c_s, \mathcal{E}_{\text{rqb}})$ given in Appendix A.6.

Revised caption c_r (teacher, high-budget view).

*The video begins with a close-up shot of two green leaves resting on a patch of soil. The soil appears to contain some dried plant matter and small pieces of wood. A few green plants are also visible in the background. A finger, likely belonging to the person holding the camera, points at the left leaf first and then the right leaf. The camera then zooms in for a closer look at the leaves, showcasing their **smooth texture** and the veins running through them. The finger continues to point at both leaves, emphasizing specific areas. The camera remains focused on the leaves throughout, with the background remaining constant. There are no other people or objects in the frame. The video seems to be a detailed examination of the leaves, possibly for educational or observational purposes. The lighting is natural, suggesting an outdoor setting. There is no audio present in the video.*

The teacher keeps the parts of c_s that the audit confirmed (e.g. “natural lighting”, “no audio”), and rewrites only the spans contradicted or underspecified by $\mathcal{E}_{\text{rqb}}$: in this example, the only edit is promoting “their texture” to “their *smooth* texture”, i.e. folding the answer of Q1 into c_r . Q2 to Q4 happen to be already implicitly consistent with c_s , while the four audio negatives Q5 to Q8 are absorbed into the unchanged closing sentence “There is no audio present in the video.”. This is the typical behavior we observe across the corpus: most QA pairs serve as silent verifications of c_s , and a small number trigger localized edits that thread the audit’s evidence into the revised text.

What Phase B sees. At training time, this single example contributes one cached tuple $(c_s, \mathcal{E}_{\text{rqb}}, c_r)$. The teacher path is teacher-forced to c_r with the

Algorithm 1 SODA: Self-Supervised On-Policy Distillation via Dual-Stream Auditing.

Require: model M_θ ; training set $\mathcal{D}$; refresh period R ; vote size K , thresholds τ, τ_{piv} ; pivot cap D_{max}

- 1: **Phase A.** For each $(V, A) \in \mathcal{D}$:
- 2: sample $c_s \sim M_\theta(V_{\text{lo}}, \text{prompt}_c)$ $\triangleright$ student draft
- 3: *Caption QA Stream:* $\{(q_i, d_i, m_i)\} \leftarrow \text{QGen}(c_s)$;
- 4: $S_{\text{skip}} \leftarrow \{d_i : q_i = \text{SKIP_NOT_IN_CAPTION}\}$
- 5: for each remaining q_i : sample K answers from $M_\theta(V_{\text{hi}}, A_{\text{hi}}, q_i)$, take majority $(\hat{a}_i, \rho_i)$
- 6: $\mathcal{E}_{\text{rqb}}^{\text{cap}} \leftarrow \{(q_i, \hat{a}_i) : \rho_i \geq \tau, \hat{a}_i \neq \text{uncertain}, d_i \notin S_{\text{skip}}\}$
- 7: *Video QA Stream (video-grounded pivot):* $U \leftarrow S_{\text{skip}} \cup \{d_i : \rho_i < \tau_{\text{piv}} \text{ or dropped}\}$, capped at D_{max}
- 8: $\{q_j\} \leftarrow \text{QGen}(V_{\text{hi}}, A_{\text{hi}}, U; \text{avoid}=\{q_i\})$
- 9: apply the same K -sample vote to each q_j to obtain $\mathcal{E}_{\text{rqb}}^{\text{vid}}$
- 10: $\mathcal{E}_{\text{rqb}} \leftarrow \mathcal{E}_{\text{rqb}}^{\text{cap}} \oplus \mathcal{E}_{\text{rqb}}^{\text{vid}}$
- 11: $c_r \leftarrow \arg \max M_\theta(V_{\text{hi}}, A_{\text{hi}}, c_s, \mathcal{E}_{\text{rqb}})$
- 12: cache $(c_s, \mathcal{E}_{\text{rqb}}, c_r)$
- 13: **Phase B.** For each step $t = 1, 2, \dots$:
- 14: sample a cached tuple $(c_s, \mathcal{E}_{\text{rqb}}, c_r)$
- 15: $\ell_s \leftarrow M_\theta(V, \text{prompt}_c; \text{forced}=c_r)$
- 16: $\ell_t \leftarrow M_\theta(V, c_s, \mathcal{E}_{\text{rqb}}; \text{forced}=c_r) \triangleright$ no gradient; same audio-visual encoding as student
- 17: update θ by minimizing Eq. 6
- 18: if $t \bmod R = 0$, re-run Phase A

textual context $(c_s, \mathcal{E}_{\text{rqb}})$ rendered as the numbered Q&A list above; the student path is teacher-forced to the same c_r but conditioned only on prompt_c . The KL in Eq. 6 is then the per-token gap between these two distributions over the caption tokens of c_r , so the supervision is densest precisely on the spans where the audit injected new content (e.g. “smooth”).

Algorithm 1 summarizes the full procedure. Phase A is executed once at the start and then every R optimization steps; between refreshes, Phase B replays the cached $(c_s, \mathcal{E}_{\text{rqb}}, c_r)$ with gradients enabled.

A.8 Qualitative Comparison: Base vs. SODA-trained Student

To complement the aggregate caption-fidelity numbers in Section 4.2, we walk through one

Omni-Cloze sample for which the SODA-trained student (frozen-teacher Round 2) writes a noticeably more faithful caption than the base Qwen2.5-Omni-7B. Concrete details that appear in the SODA caption *and* also surface in the ground-truth Omni-Cloze passage but are missing from the base caption are highlighted in **green**. We deliberately keep this comparison qualitative; the corresponding aggregate numbers are reported in Table 2.

Sample: Comedic monologue in a living room.

A woman delivers a sketch monologue while a dog yawns on the couch behind her; bold captions appear in sync with her speech.

Base caption.

The video opens with a woman standing in a cozy living room. ... A coffee table sits in front of the couch, holding a few books and a small potted plant. To the right of the couch, there is a bookshelf filled with books and some decorative items. ... As she walks towards the camera, she starts talking. The on-screen text appears, reading "OH, SORRY. DID YOU EXPECT ME TO FLINCH? TO HAVE A REACTION?" ... a large black and brown dog is seen lying on the couch. ... The audio quality is clear, allowing the viewer to easily understand the woman's speech and the on-screen text.

SODA caption.

The video opens with a woman standing in a living room. ... A small table in front of the couch holds a few items, including a cup and a book. A bookshelf filled with books is visible in the background, along with a framed picture on the wall. A large dog is lying on the couch, looking relaxed. ... The on-screen text appears in yellow and white, saying "OH, SORRY. DID YOU EXPECT ME TO FLINCH. TO HAVE A REACTION? ...".

The SODA caption recovers a cup on the coffee table, a framed picture on the wall, and crucially the colour scheme of the on-screen captions (yellow and white), all of which are anchored as cloze blanks in the ground-truth passage but absent from the base caption. The SODA caption is also more cautious about the dog: it reports a large dog *looking relaxed* rather than committing to the unverified *black and brown* colour that the base caption invents.

A.9 Extended Discussion of Related Work

This appendix gives the longer version of the three threads briefly summarized in Section 2.

A.9.1 On-Policy Distillation

Knowledge distillation (Hinton et al., 2015) trains a student to imitate a teacher’s output distribution, and has long been a backbone of compressing language models. Sequence-level distillation (Kim and Rush, 2016) extends this idea to autoregressive generation by aligning whole-sequence likelihoods, while f -divergence variants (Wen et al., 2023; Gu et al., 2024) replace the forward-KL with reverse-KL or Jensen-Shannon to reduce mode-averaging in language generation. A central problem in this line is the *train-inference mismatch*: distillation traditionally fits the student’s distribution at the teacher’s outputs, while at deployment the student decodes from its own. On-policy distillation (Agarwal et al., 2024; Ko et al., 2024) addresses this by sampling from the student during training and computing the teacher’s likelihood on those samples, so that the supervision signal lives on the trajectories the student actually visits. G-OPD (Yang et al., 2026) ports this paradigm to multimodal LLMs by using a stronger external text-only model as the teacher and aligning student outputs along the text channel.

This entire family relies on a teacher that is *stronger* than the student in the standard sense: more parameters, data, or compute. SODA explores a different point: the teacher and student share weights, and the teacher’s edge comes from a richer input rather than a larger model, so the supervision reflects an information gap rather than a capacity gap and removes the dependence on an external stronger model.

A.9.2 Self-Distillation and Self-Training

A separate line of work uses a model to teach itself. Born-again networks (Furlanello et al., 2018) retrain a student of the same architecture on the previous generation’s outputs and observe consistent gains. Be-Your-Own-Teacher (Zhang et al., 2019) distills deep layers into shallow ones inside a single network. In semi-supervised learning, mean teachers (Tarvainen and Valpola, 2017) and noisy student (Xie et al., 2020) construct the teacher from a stable copy of the student plus stronger augmentations on the unlabeled side. For language reasoning, STaR (Zelikman et al., 2022) and ReST (Singh et al., 2024) bootstrap a model by sampling rationales or solutions from itself, filtering with a correctness oracle, and fine-tuning on the survivors; self-consistency (Wang et al., 2023) aggregates multiple samples into a single, more

reliable answer. Recent video-side work such as Video-STaR (Zohar et al., 2025) extends this self-training idea to instruction tuning.

These methods typically separate teacher and student along a *noise* or *stochasticity* axis (augmentations, EMA, majority voting) and supervise on the final answer, which provides only sparse learning signal. SODA instead separates them along an *information* axis, where the teacher sees a higher-fidelity audio-visual view together with the student’s draft and a model-produced Revision QA Bank, and supervises on caption tokens, turning text-side and multimodal-side input-budget lifts into a dense token-level signal for an omni-modal VideoLLM.

A.9.3 Reducing Hallucinations in Video and Multimodal LLMs

Hallucination is a recognized failure mode of multimodal LLMs and has been studied along several axes. On the evaluation side, POPE (Li et al., 2023) and HallusionBench (Guan et al., 2024) target object-level and entanglement-level hallucinations in image LLMs, and VideoHalluciner (Wang et al., 2024), VidHalluc (Li et al., 2025a), EventHallusion (Zhang et al., 2025b), and AVHBench (Sung-Bin et al., 2025) extend the diagnosis to videos and to audio-visual interactions. On the mitigation side, three families dominate. *Decoding-time interventions* such as VCD (Leng et al., 2024) and OPERA (Huang et al., 2024) contrast the model’s distribution under perturbed inputs to suppress over-confident hallucinations. *Preference-based fine-tuning* methods including RLHF-V (Yu et al., 2024), POVID (Zhou et al., 2024a), HA-DPO (Zhao et al., 2024), and LURE (Zhou et al., 2024b) collect or synthesize hallucination-free vs. hallucinated pairs and apply DPO (Rafailov et al., 2023) or RLHF on top of the answer tokens. *Reinforcement-style methods* such as SDRL (Wang et al., 2026) and VideoHallu (Li et al., 2025b) optimize answer accuracy or hallucination metrics with GRPO (Shao et al., 2024; Schulman et al., 2017). A complementary thread uses an external process or reward model, e.g. VisualPRM (Wang et al., 2025a), to score intermediate reasoning steps.

These methods place supervision on answer tokens (or contrastive perturbations of them), often require curated preference data, reward models, or an RL loop on top of SFT, and provide a sparse signal, namely a binary label or a scalar reward

per trajectory. SODA differs on all three axes: a dense token-level KL on the revised caption, a same-weight teacher with no preference data or reward model, and audio and video input-budget lifts treated symmetrically inside the teacher’s input; we therefore view our gains on hallucination benchmarks as complementary to this literature rather than competing with it.